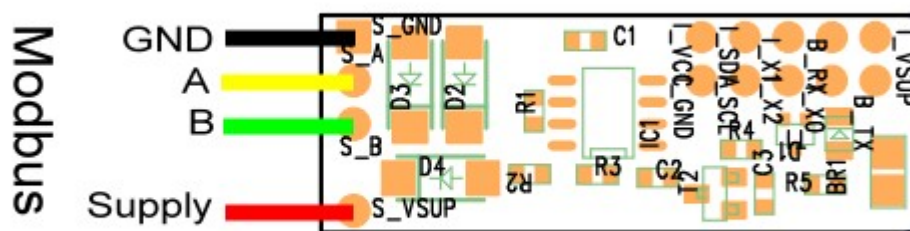


OSX – Open-SDI12-Blue Bluetooth®

Version Modbus Konverter, Type 210

1 Setup

Mit dem OSX Modbus Konverter können Modbus Sensoren an einem SDI-12 Bus betrieben werden:



Die Konverterplatine ist auf der Rückseite eines OSX Knotens (basierend auf der Open-SDI12-Blue-Plattform) montiert. Stromversorgung des Modbus erfolgt über die SDI-12 Schnittstelle und kann vom Konverter geschaltet werden. Der Konverter kann bis zu 10 Register des Modbus verwalten.

SDI-12-Kabel (Aderenden):

SCHWARZ: GND
 BRAUN: 6-V-16-V-Versorgung
 WEISS: SDI-12-Signal

Der Befehlssatz basiert auf dem Standard-SDI-12-Befehlssatz (V1.3). Die wichtigsten Befehle:

aAn!: Adresse von „a“ in „n“ ändern. (a kann immer ein „?“ als Platzhalter sein.)

aI!: Knoten identifizieren (sollte als „a13TT_MOB_A_0390_OSXxxxxxxxx“ identifiziert werden.)

aM!: Messung starten (auch „aMC!“ mit CRC16). Dadurch wird die Messung gestartet. Nach Abschluss stehen alle Messwerte in einem internen Cache zur Verfügung. Mit dem „D“-Befehl können bis zu 9 Ergebnis-Register ausgelesen werden.

aM1!: Messung starten (auch „aMC1!“ mit CRC16). Die Messung inklusive Versorgungsspannung wird gestartet. Nach Abschluss stehen alle Messwerte im internen Cache zur Verfügung. Mit dem „D“-Befehl können bis zu 9 Ergebnis-Register plus Versorgungsspannung ausgelesen werden.

aD0!: Die ersten der maximal 10 Ergebnis-Register des vorhergehenden „M“-Befehls werden ausgegeben. Weitere Register gegebenenfalls mit „aDn“, n=0-9.

Fehlercodes (alle Werte kleiner gleich -700.000):

-700..-755: MODBUS Error Codes (700: Code 0..755: Code:255)
-1000: Interner Sensorfehler („Keine Antwort“) (wahrscheinlich Sensor oder interne Verbindung unterbrochen)
Sonstige: Anzeige als Text in BLX.JS oder BlueShell

2 Modbus – Kurzübersicht

Der Modbus verwendet RS485. Damit lassen sich zwar größere Entfernungen als wie mit SDI-12 überbrücken (bis zu mehrere km sind möglich), aber Modbus verwendet mindestens 4 Adern und Modbus-Sensoren können nicht so einfach schlafen. Daher kann der Konverter auch bei Bedarf den ganzen Modbus dynamisch versorgen.

Üblicherweise werden für Modbus-Sensor oft höhere Spannungen als für SDI-12 benötigt. Meist ab 5V, oft aber mindestens 8V oder mehr.

Jeder Modbus-Sensor hat eine fest vergebene, eigene Adresse.

Die Kommunikation erfolgt üblicherweise im Format 9600 Baud, 1 Start- und 1 Stop-Bit, keine Parität („9600 8N1“). Bei anderen Baudraten bitte gegebenenfalls bei uns nachfragen!

2.1 Modbus-Register

Ein Modbus-Sensor kann 4 verschiedene Registergruppen enthalten:

- 0, 1: Diskrete Ein- und Ausgänge
- 3, 4: 16-Bit-Register

Wichtig: Der (logischen) Registerbezeichnung steht immer die Gruppe voran. Innerhalb jeder Gruppe sind 65536 Register möglich, sie werden von 1 – 65536 durchnummeriert. Da binär ab 0 gezählt wird, werden sie (physikalisch) ab 0 indiziert. Der (logischen) Registerbezeichnung von z.B. 3010 entspricht also der Index 9 (= 10-1) in Gruppe 3. Und z.B. logisch 311, 3011, 30011 sind die Namen des Folgeregister, die ‚0‘ zählt nicht.

Es können mehrere physikalische Register zusammengefasst werden (z.B. 2 fortlaufende 16-Bit-Register können einen Fließkommawert darstellen). Das Zahlenformat ist immer „Big Endian“.

2.2 Setup-Parameter

In der Regel liegen den Sensoren Details zu den Adressen und Register bei. Zum Testen des Protokolls verfügt der Konverter über ein spezielles Debug-Kommando.

Sehr bequem für bekannte Sensoren ist das Setup mittels QR-Code (siehe nachfolgend ‚.crun‘).

3 Die Open-SDI12-Blue-Plattform

OSX-Sensoren basieren auf einer offenen Plattform:

Link: <https://github.com/joembedded/Open-SDI12-Blue>

4 Software

4.1 Software für den Sensorzugriff

OSX-Sensoren können über SDI-12 (V1.3) oder Bluetooth BLE oder SDI12 über Bluetooth angesprochen werden.

- BlueShell für PC (Windows 10/11)
- BLX.JS / BlxDashboard (PC (Browser: Chrome, Edge, Opera, ...) oder Android). Keine App erforderlich! Dieses Tool ist komplett in Javascript geschrieben. Quellcodes für eigene Anpassungen stellen wir gerne zur Verfügung.

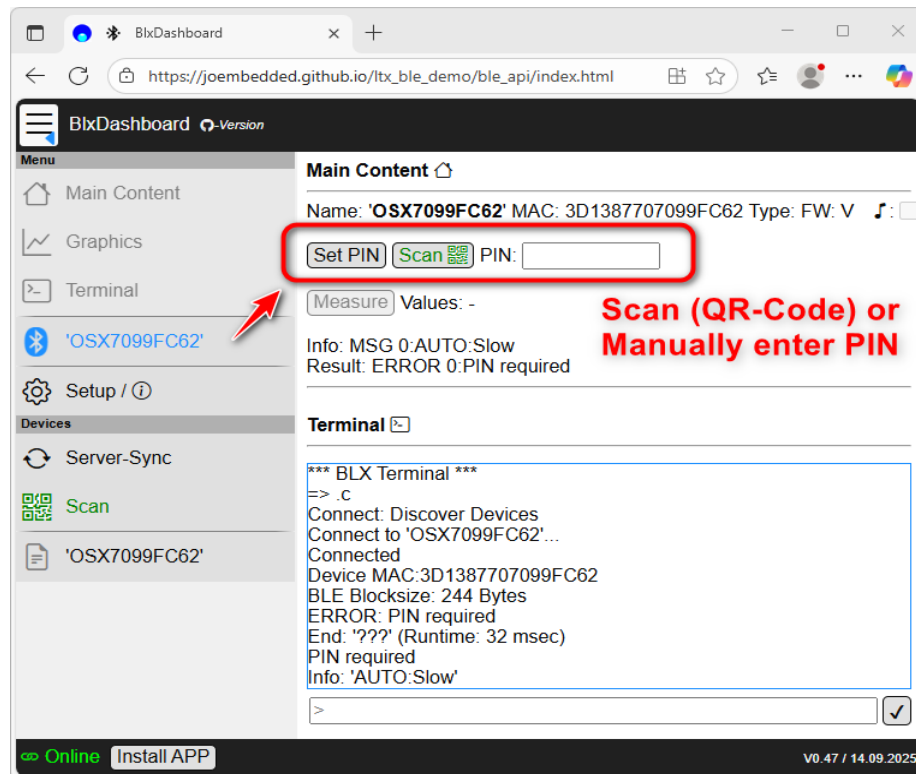
Link: <https://joembedded.de/x3/blueshell/>

4.2 Software für SDI12

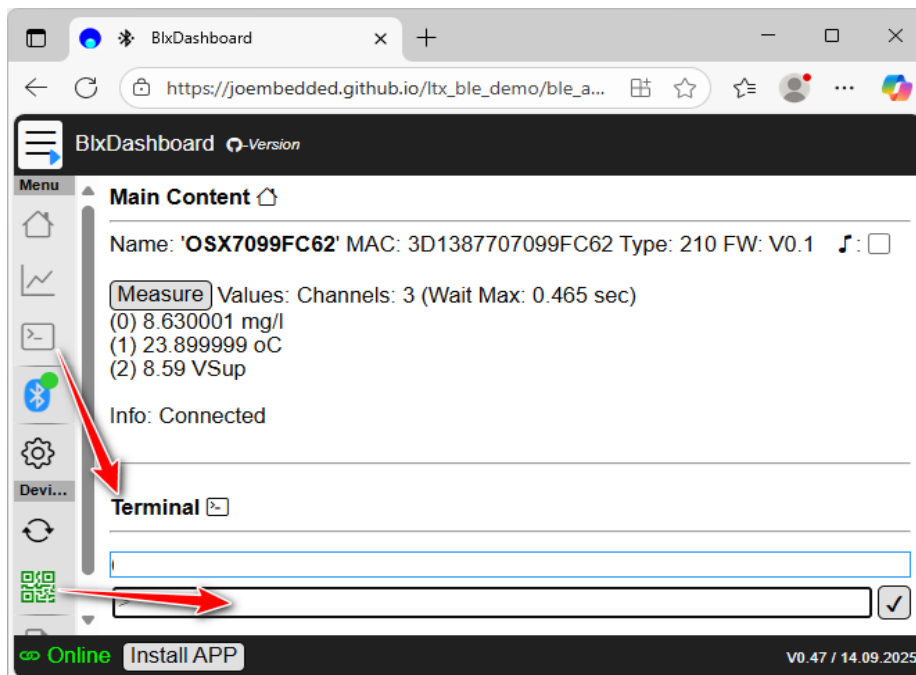
- SDI12Term: Ein einfaches Terminal für PC (Windows) (Anschluss von SDI12-Sensoren über RS232). Über SDI12Term kann einfach via SDI-12 kommuniziert werden. Quellcodes sind frei verfügbar, geschrieben in C.

Link: <https://github.com/joembedded/SDI12Term>

5 Beispielsitzung mit dem BLX Dashboard



Die Geräte sind mit einer 6-16 stelligen PIN versehen. Dieser liegt als QR-Code oder Klartext bei und muss entweder (einmalig) gescannt oder manuell eingegeben werden (Die Authentifizierung erfolgt nach der Challenge-Response-Methode, d.h. die PIN wird niemals im Klartext übertragen).



Terminal-Kommandos können manuell oder per QR-Code gesendet werden.

6 Kommandos

Hier eine kurze Übersicht zu den wichtigsten Funktionen.

6.1.1 Kommandos für diesen (Type 210):

Messen:

- M oder MC oder M1 oder MC1 startet eine Messung
- D liefert die Ergebnisse

6.1.2 Standard Kommandos Open-SDI12-Blue („SDI-12 via BLE“):

Über Bluetooth können auch reguläre SDI-12-Kommandos gesendet werden. Dazu muss einfach ein „z“ vorangestellt werden:

> z?I!	Identify Sensor
Reply: '013TT_MOB_A_0390_OSX7099FC62<CR><LF>'	
End: 'OK' (Runtime: 188 msec)	
> z?M!	Measure
Reply: '00012<CR><LF>'	
Reply: '0<CR><LF>'	
End: 'OK' (Runtime: 269 msec)	
> z?D0!	Get Results
Reply: '0+8.550001+24<CR><LF>'	
End: 'OK' (Runtime: 315 msec)	
> z?MC!	Measure (with CRC)
Reply: '00012<CR><LF>'	
Reply: '0<CR><LF>'	
End: 'OK' (Runtime: 205 msec)	
> z?D0!	Get Results
Reply: '0+8.54+24JqN<CR><LF>'	(,JqN' = CRC)
End: 'OK' (Runtime: 318 msec)	
> z?XDevice!	Get Details and PIN via SDI-12
Reply: '0M:3D1387707099FC62,T:210,V0.1,P:145540!<CR><LF>'	
End: 'OK' (Runtime: 171 msec)	
> z?XFactoryReset!	SDI-12: Factors Reset
Disconnected (FactoryReset)	

6.1.3 Zusätzliche Kommandos

Einige wichtige weitere Kommandos:

> .a	.a or .audio: Audio
Audio: RSSI: OFF, Term: ON	and „Finder♫“
> .audio 1 1	Audio & Finder♫ ,ON'
Audio: RSSI: ON, Term: ON	
> .firmware	Secure firmware
Select new firmware (*.sec)...	update
> bNEW_NAME	Set new BLE
advertising Name	
End: 'OK' (Runtime: 10 msec)	
> k	List Coefficients
Reply: 'K0: 0.010000 'mg/l'.Multi'	
Reply: 'K1: 0.000000 'mg/l'.Offset'	
Reply: 'K2: 0.100000 'oC'.Multi'	
Reply: 'K3: 0.000000 'oC'.Offset'	
Reply: 'K4: 1.000000 '#2'.Multi'	
Reply: 'K5: 0.000000 '#2'.Offset'	
...	
Reply: 'C:'r55 0 IsIs''	
Reply: 'U:'mg/l oC''	
Reply: 'P(msec):0'	
End: 'OK' (Runtime: 857 msec)	
> z?XK0!	Query Coefficient K0
Reply: '0K0=0.010000<CR><LF>'	
End: 'OK' (Runtime: 250 msec)	
> z?XK1=-0.123!	Modify Coefficient K1
Reply: '0K1=-0.123000<CR><LF>'	
End: 'OK' (Runtime: 179 msec)	
> z?XC=r0x14 0x11 IssI!	Modify Modbus Command
Reply: '0C='r0x14 0x11 IssI'<CR><LF>'	
End: 'OK' (Runtime: 213 msec)	
> z?XU=oC mg/l!	Modify Units
Reply: '0U='oC mg/l'<CR><LF>'	
End: 'OK' (Runtime: 464 msec)	
> z?XP=5000!	Modify Power-Up to
5000 msec	
Reply: '0P=5000<CR><LF>'	
End: 'OK' (Runtime: 248 msec)	
> z?XWrite!	Save new Coefficients
Reply: '0<CR><LF>'	to Flash
End: 'OK' (Runtime: 230 msec)	

Advertising Name

Das Kommando ,b‘ erlaubt es, dem Sensor einen neuen Bluetooth-Namen zu geben (3 – 11 Zeichen).

Koeffizienten und Messparameter

Hinweis: Diese Werte lassen sich auch über SDI-12 setzen (,z‘):

Der Sensor hat für jeden der 8 Ergebnis-Register 2 Koeffizienten: Steilheit (Multi, Default: 1.0) und Offset (Default: 0.0):

Die Berechnung erfolgt nach:

$$\text{WERT} = (\text{GEMESSEN} * \text{Multi}) - \text{Offset}$$

Das **Messkommando ,C‘** liest 1-n Modbus-Register aus:

,r‘ steht für Gruppe 3

,h‘ steht für Gruppe 4

danach folgt die Modbus-Adresse (dezimal oder als 0x für Hex)

danach die auszulesenden Register:

I 2-Byte Integer-Zahl (-32768...+32767) (= 1 Modbus-Register)

F 32-Bit Fließkommazahl (= 2 Modbus-Register)

s ,skip‘ – Ignoriere 2 Bytes

S ,Skip‘ – Ignoriere 4 Bytes

Trenner ist jeweils ein Leerzeichen, mehrere Messkommandos in Folge sind erlaubt, z.B:

,r16 II r0x55 FssF‘ liest 2 Integer und 2 Fließkomma-Werte von 2 Modbus-Sensoren.

Das **Kommando ,U‘** legt die Einheiten fest (nur für Bluetooth-Messung): 1-7 Zeichen pro Einheit, Trenner auch hier Leerzeichen.

Das **Kommando ,P‘** (in msec) legt fest, ob der Modbus dauerhaft versorgt wird (wenn 0) oder wie lange vorher er eingeschaltet werden soll („WarmUp“).

Sämtliche Änderungen werden erst bei „z?Xwrite!“ gespeichert.

Das **Kommando ,pdebug‘** aktiviert temporär die Ausgabe der Modbus-Daten, beispielsweise um das korrekt ,C‘-Kommando zu finden. Nur ,p‘ deaktiviert Debug:

Reply: 'TX> 37 3 0 0 0 4 41 9F'

37:Adr, 3:Fkt ... CRC: 419F

Reply: 'RX< 37 3 8 3 70 0 2 0 EE 0 1 9D B8'

6.1.4 Schnelles Setup mit ‚.crun‘

Für bekannte Sensoren stehen die benötigten Kommandos als Textdatei oder -noch bequemer- zum Scannen per QR-Code zur Verfügung.

Dazu entweder im BlxDashboard den QR-Code-Scanner aktivieren:

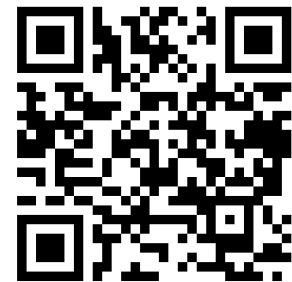


Oder in der Terminal-Kommandozeile das entsprechende Kommando eingeben.

- *Dragino DR-DO1 Dissolved Oxygen Sensor*

Ein einfacher Sensor für gelösten Sauerstoff auf Fluoreszenzbasis

Kommando: `.crun crun/0210_modbus_draginoo2.crun`



- *SeedStudio MODBUS Dissolved Oxygen Sensor*

Ein einfacher Sensor für gelösten Sauerstoff auf Fluoreszenzbasis

Kommando: `.crun crun/0210_modbus_seedo2.crun`



7 Stromversorgung

Der OSX-Sensor arbeitet von 2,8 V bis 16 V (siehe Open-SDI12-Blue-Dokumentation).

Für Modbus sind jedoch meist mindestens 5 V oder mehr erforderlich, empfohlen: 8 V–16 V, siehe Doku zum jeweiligen Modbus-Sensor

Für die Messung: max. 100 mA vom Konverter schaltbar

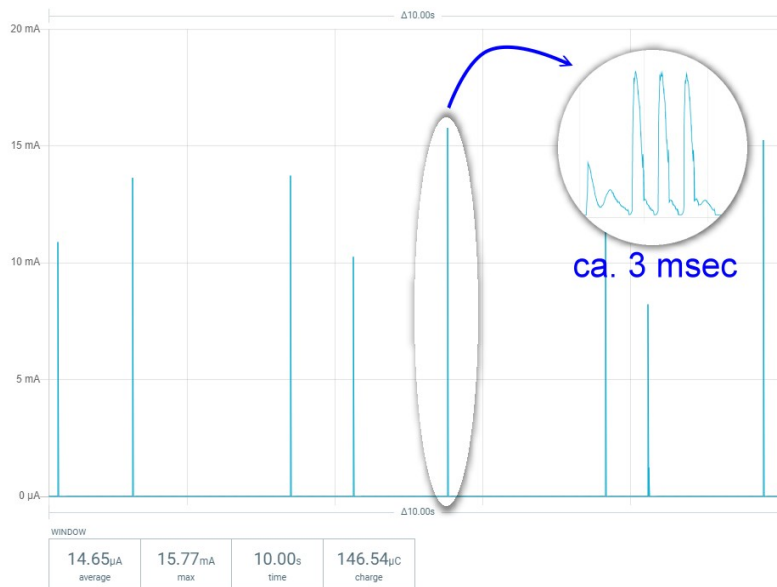
Betriebstemperatur: -40 °C–+85 °C.

7.1 Power Profile

7.1.1 Power Up Sequence

The Sensor is ready after ca. 250 msec.

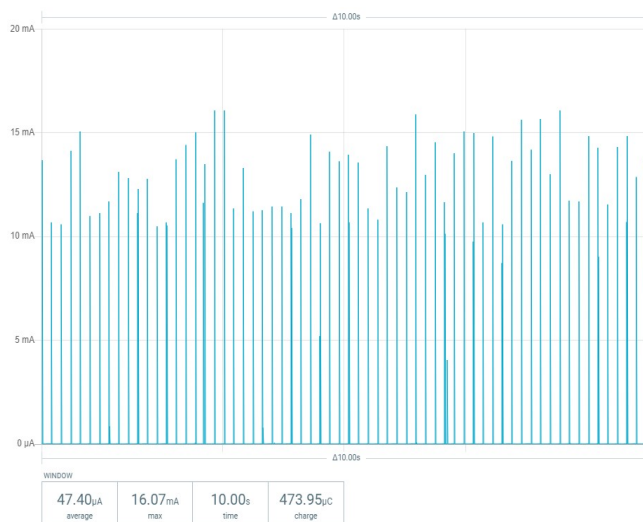
7.1.2 Advertising (in deep sleep)



Advertising power consumption (one peak zoomed)

Average power consumption in deep sleep is $<15 \mu\text{A}$ @ 4V

7.2 Connected Mode



Connected power consumption

In Connected Mode (active BLE connection) the average power consumption is $<50 \mu\text{A}$ @ 4V

8 Compliance (Version: Modbus Converter)

8.1 Compliance: CE, RoHS



- EN 55022 Emission, class B < 30 dB μ V/m (0.03...1 GHz)
- EN 61000-4-2 Electrostatic discharge 4 kV contact / 8 kV air
- EN 61000-4-3 Irradiated RF 10V/m (0.1...1 GHz)
- EN 61000-4-4 Transients (burst) 4 kV
- EN 301 489-1 V2.1.1 and EN 301 489-17 V3.1.1 EMC
- EN 300 328 V2.1.1 EN 300 330 V2.1.1 Radio Emission
- Bluetooth SIG listed: ID 138612

The sensor OSX – Version Modbus Converter, Type 210 complies with the essential requirements of Radio Equipment Directive (RED) 2014/53/EU and with the Directive 2011/65/EU (EU RoHS 2) and its amendment Directive (EU) 2015/863 (EU RoHS 3).

Manufacturers:

GeoPrecision GmbH
Am Dickhäuterplatz 8
D-76275 Ettlingen

Terratransfer GmbH
Ottostr. 19a
D-44867 Bochum

15.09.2025

A handwritten signature in black ink, appearing to read 'Jürgen Wickenhäuser'.

Jürgen Wickenhäuser (R&D)
